...e costruisci il tuo LABORATORIO DIGITALE

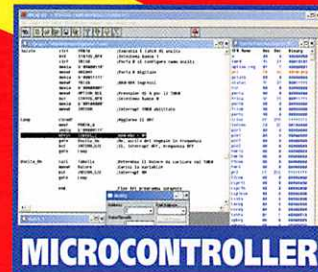
6,90 €

60

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41060

Peruzzo & C.

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

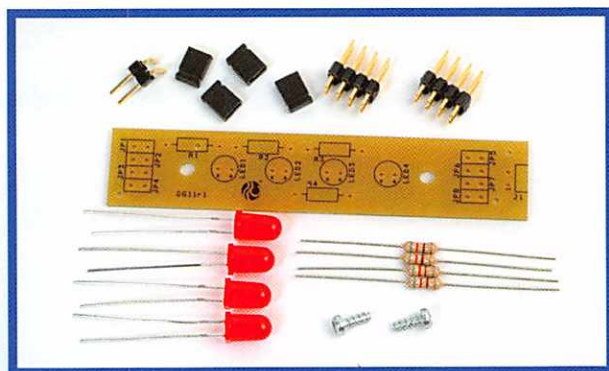
Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

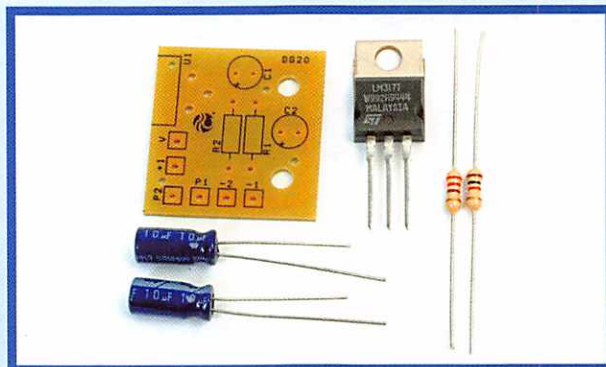
"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

impara l'elettronica digitale

IN REGALO in questo fascicolo

1 Scheda DG11r1
4 LED rossi
2 Connettori maschio da c.s. da 2x4 vie diritti
1 Connettore maschio da c.s. a 90° a 2 vie
4 Resistenze da 820 Ω 5% 1/4 W
4 Ponticelli isolati neri
2 Viti con filetto per plastica

IN REGALO nel prossimo fascicolo

1 Scheda DG20
1 Circuito integrato LM317T
1 Resistenza da 220 Ω 5% 1/4 W
1 Resistenza da 10 K 5% 1/4 W
2 Condensatori elettrolitici da 10 µF

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

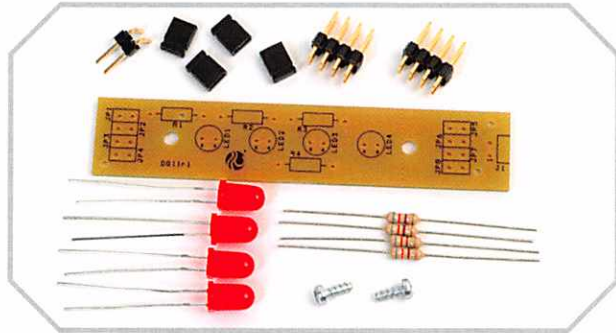
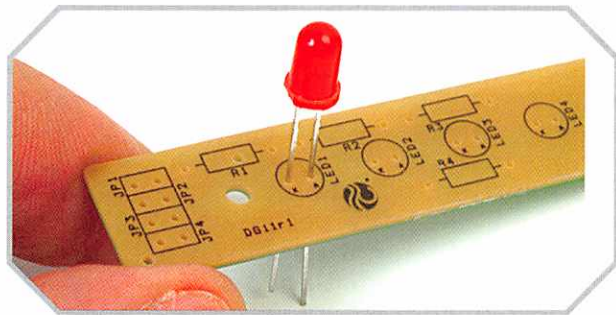
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail: elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

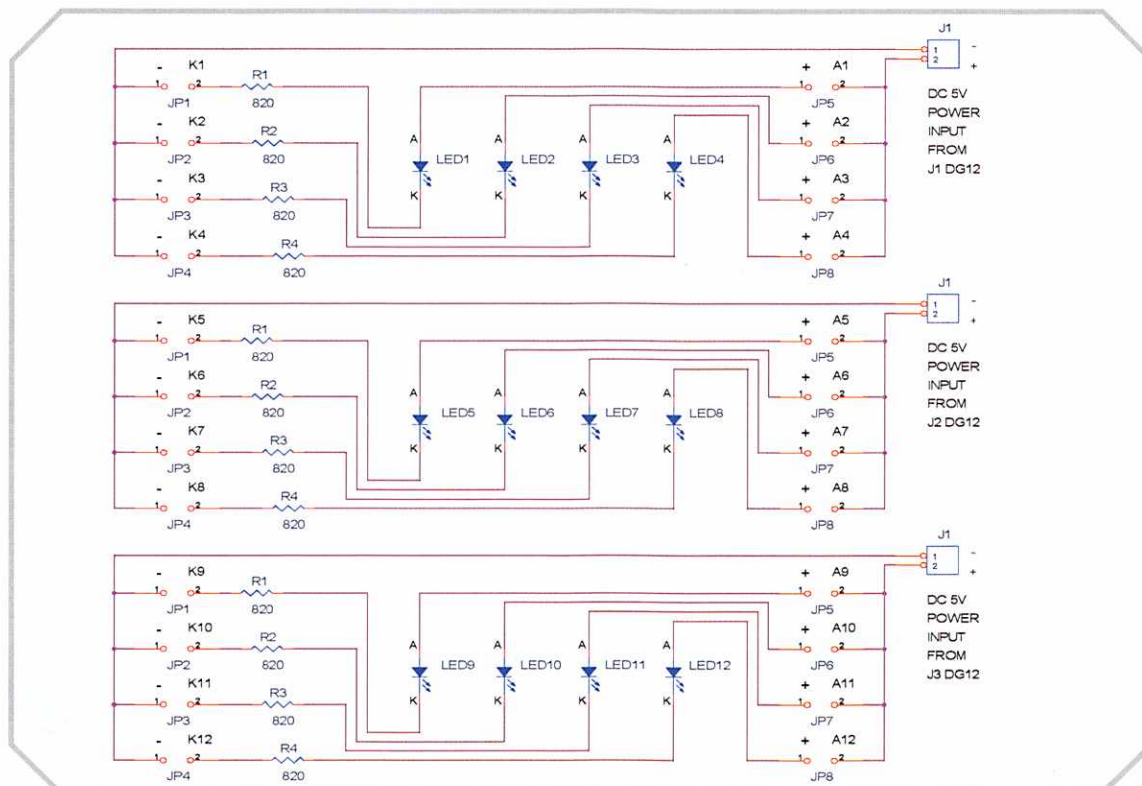
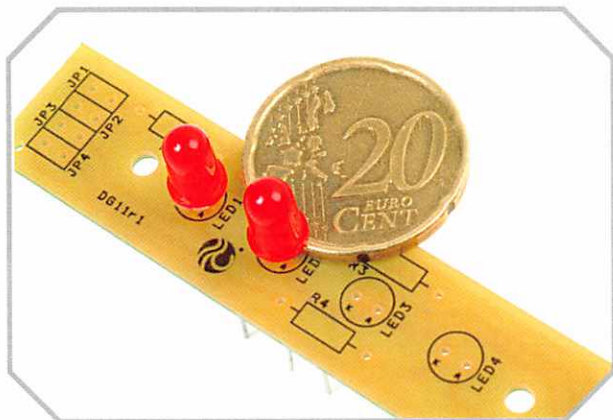
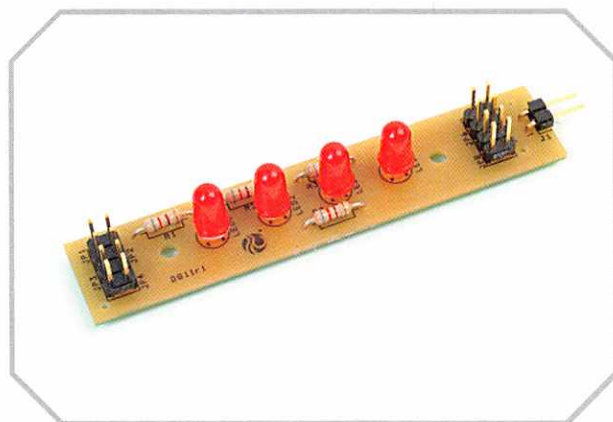
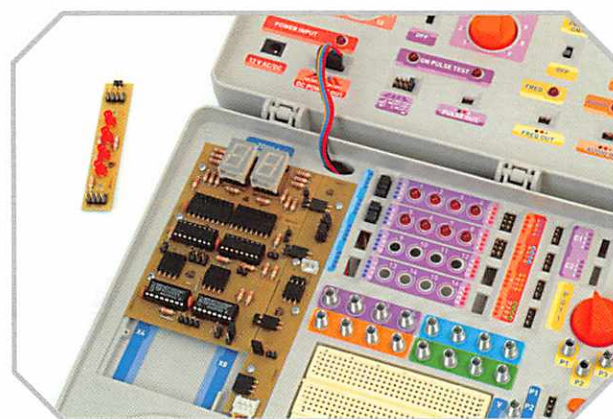
Terzo modulo della matrice dei LED

Materiale allegato a questo fascicolo.

Attenzione alla polarità dei LED.

Con questo fascicolo si forniscono tutti i componenti necessari per il montaggio e l'installazione del terzo circuito DG11, in questo modo aggiungendo una terza fila, avremo 12 LED nella matrice.

Il materiale

Abbiamo ora a disposizione un altro circuito stampato, quattro LED ad alta luminosità da 5 mm di diametro di colore rosso, due connettori maschio diritti a due file da quattro vie, un connettore maschio a 90° a due vie e quattro resistenze da 820 Ω. I quattro ponticelli si possono utilizzare su questa scheda o su altre posizioni in cui siano necessari per eseguire degli esperimenti.

Il potenziometro

Il montaggio dei componenti di questo circuito è stato spiegato dettagliatamente nella scheda Hardware 33 del fascicolo 17, tuttavia

Schema elettrico.

Possiamo utilizzare una moneta come separatore.

Circuito stampato DG11 completato.

Zona dove bisogna installare la terza fila dei LED.

anche se abbiamo già ripetuto il montaggio due volte, è consigliabile eseguire il lavoro attentamente per evitare errori di montaggio.

I primi componenti da installare sono i quattro LED, che hanno polarità: il terminale del catodo è quello più vicino alla zona piatta del contenitore del LED e si inserisce nel foro indicato con la lettera K sul circuito stampato.

È necessario anche fare attenzione all'altezza dei LED che deve essere uguale per tutti con un distacco di circa 2 mm dalla scheda. Possiamo utilizzare una moneta come separatore durante la saldatura, così come mostrato nella fotografia.

Saldiamo per primo uno dei terminali, dopodiché verificheremo che ogni LED sia perfettamente verticale, in caso contrario ne correggeremo la posizione prima di saldare il secondo terminale. Le resistenze, tutte da 820 Ω (grigio, rosso, marrone) si saldano secondo la procedura abituale. I terminali devono rimanere ben allineati e perfettamente verticali, dato che ospiteranno i ponticelli per l'alimentazione degli anodi o dei catodi, in base alle esigenze di ogni esperimento. Dopo aver terminato il montaggio dei componenti si verificano tutte le saldature e si controlla nuovamente la polarità dei LED.

Preparazione del montaggio

Prima di installare questo modulo sul laboratorio è necessario fare un piccolo lavoro, che vi spieghiamo di seguito:

– Per prima cosa scollegheremo tutte le alimentazioni, togliendo anche le pile dai portabatteria.

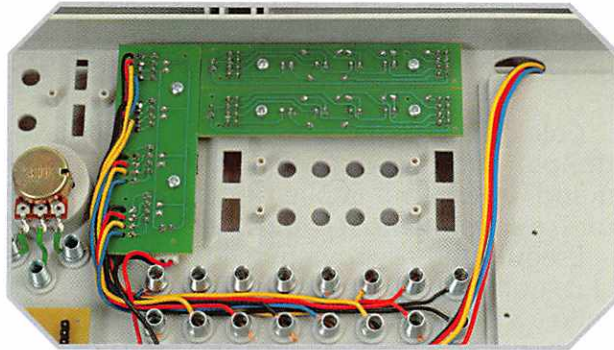
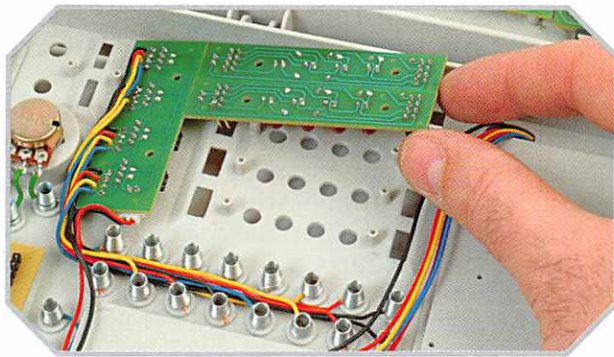
– Togliamo le viti che fissano le due schede DG11 già montate e quelle che fissano la DG12.

– Si tolgono le schede DG11 dal lato opposto alla DG12, facendo attenzione ad alzarle il minimo possibile. In realtà è sufficiente alzare solamente il lato delle schede DG11.

Installazione del terzo modulo

Per installare la terza scheda DG11 si eseguono le seguenti operazioni:

– Si inserisce la terza scheda DG11 vicino alle precedenti e con la stessa inclinazione, collegando J1 su J3 della DG12. Le quattro schede devono formare un piano.

– Si abbassa poco a poco l'insieme delle

Interno del laboratorio.

Si tolgono le viti e si sollevano le schede da un lato.

quattro schede, in modo graduale e facendo attenzione che i 12 LED e i loro connettori fuoriescano dal pannello frontale.

– Dopo che le schede saranno state appoggiate verificheremo che dall'altro lato fuoriescano i LED dall' 1 al 12.

– Si montano le viti di tutte le schede ma senza stringere a fondo.

– Inseriamo almeno un ponticello su ogni connettore per centrare le schede.

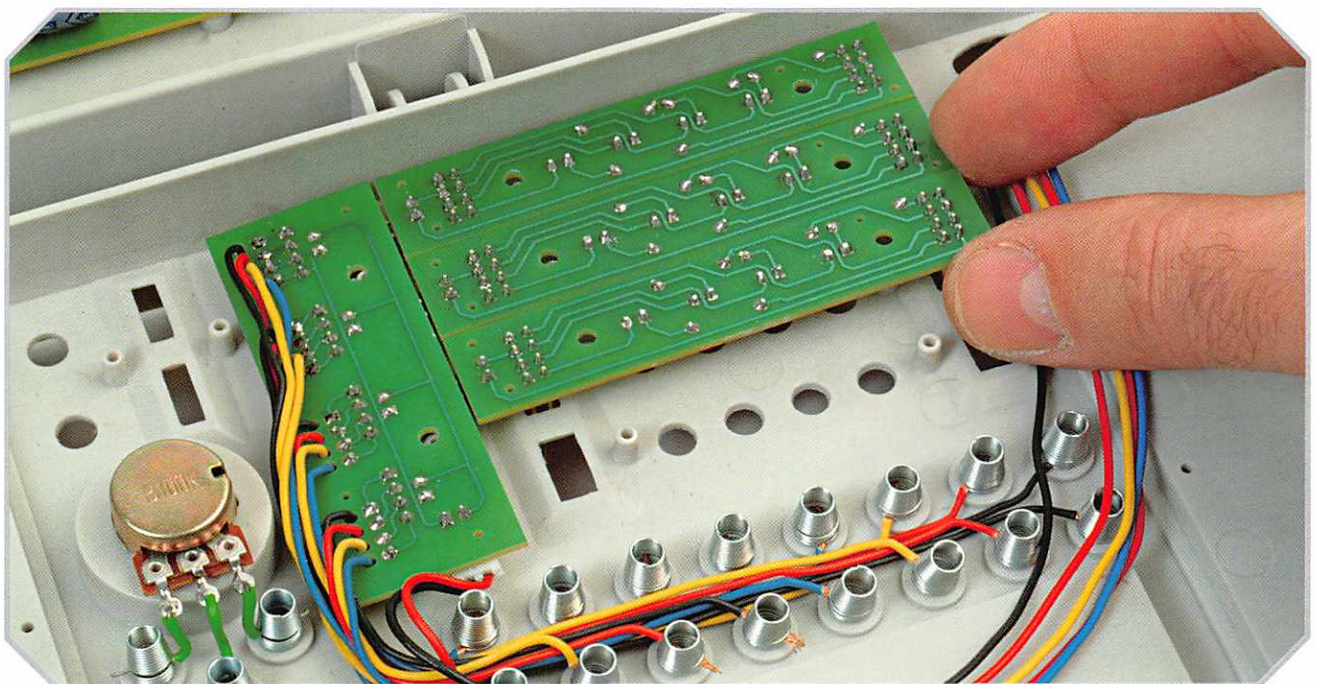
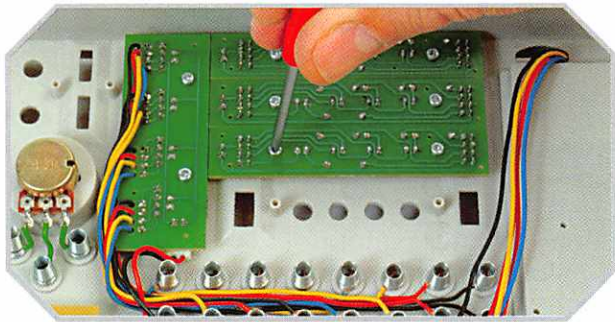
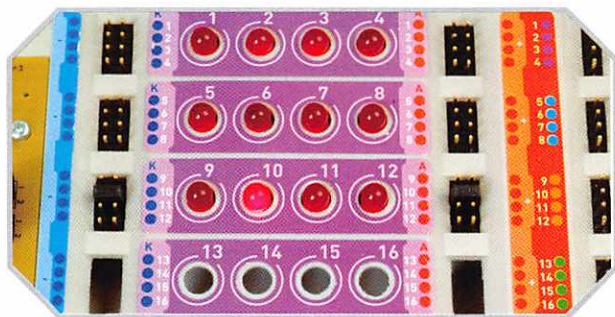
– Avvitiamo a fondo le viti senza stringere troppo per evitare di danneggiare i filetti.

Identificazione

I circuiti DG11 sono tutti uguali, quindi i loro riferimenti di montaggio sono identici, i LED sono indicati da LED 1 a LED 4, ma questi riferimenti rimangono nascosti e gli schemi utilizzano riferimenti riportati sul pannello del laboratorio:

da LED 1 a LED 4 sulla prima fila, da LED 5 a LED 8 sulla seconda e da LED 9 a LED 12 sulla terza.

Prova

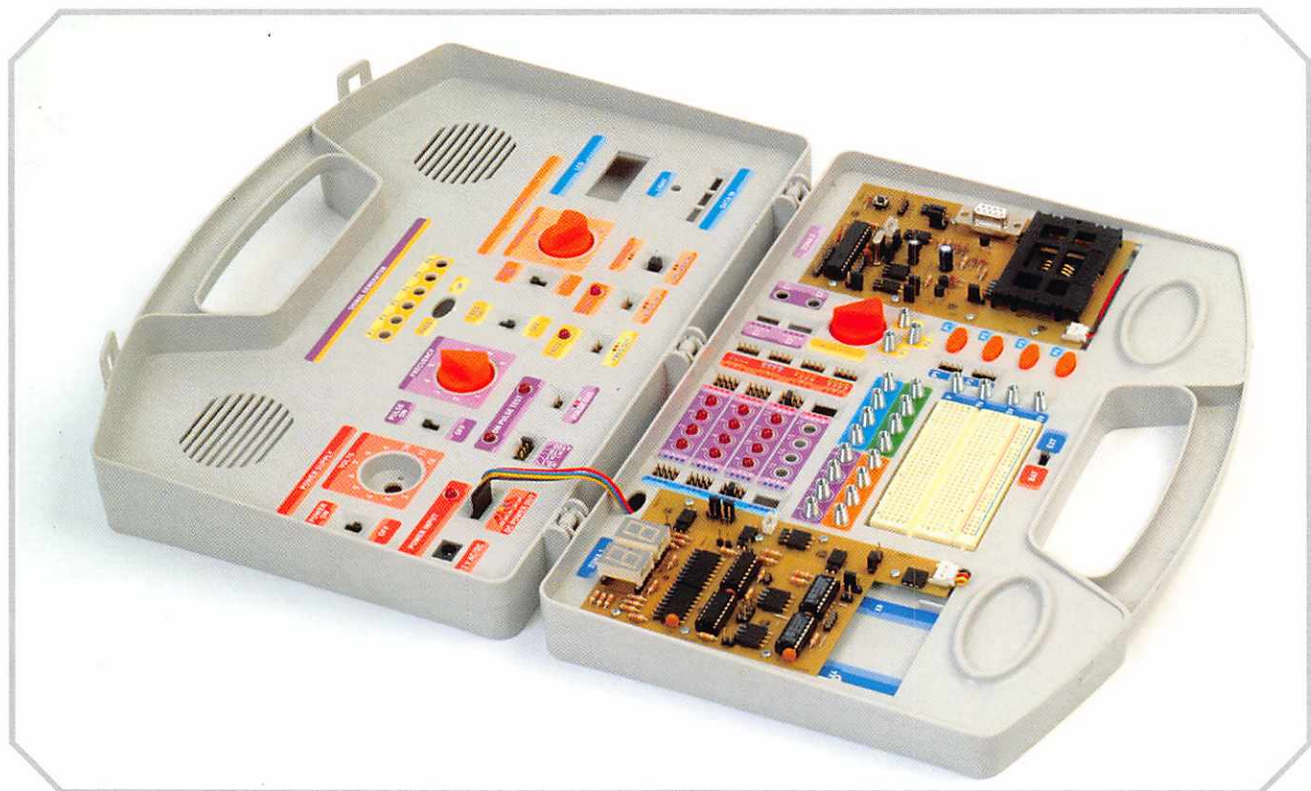
Per verificare il lavoro svolto procederemo come per le due precedenti schede.

Collegamento della terza scheda.

Fissaggio delle schede.

I LED si provano individualmente.

Nessun ponticello deve essere inserito sui connettori maschi a due file da quattro vie, posizionati su entrambi i lati delle due file dei LED.

Collegheremo l'alimentazione al laboratorio, per questa prova sono sufficienti 5 volt, quindi inseriremo le pile nel primo portabatterie e sposteremo il commutatore su BAT.

Eseguiremo la prova individuale di ciascun LED, quindi sarà sufficiente inserire un ponticello tra K1 e il terminale (-) al suo fianco e l'altro ponticello tra A1 e il terminale (+) al suo fianco, sullo stesso connettore; se tutto è corretto e le batterie sono cariche il LED 1 si deve illuminare.

Toglieremo poi i ponticelli e li porteremo sul LED 2 per verificarne il funzionamento, e così via fino a completare i 12 LED montati.

Se qualche LED non si illumina, per prima cosa verificheremo l'alimentazione, successivamente che i connettori della scheda DG11 siano ben collegati sulla scheda DG12. Se questo non risolve il problema è necessario verificare le saldature, oltre a controllare che i LED siano stati montati sulla scheda con la polarità corretta.

Laboratorio con la terza fila di LED installata.

Generatore di due frequenze

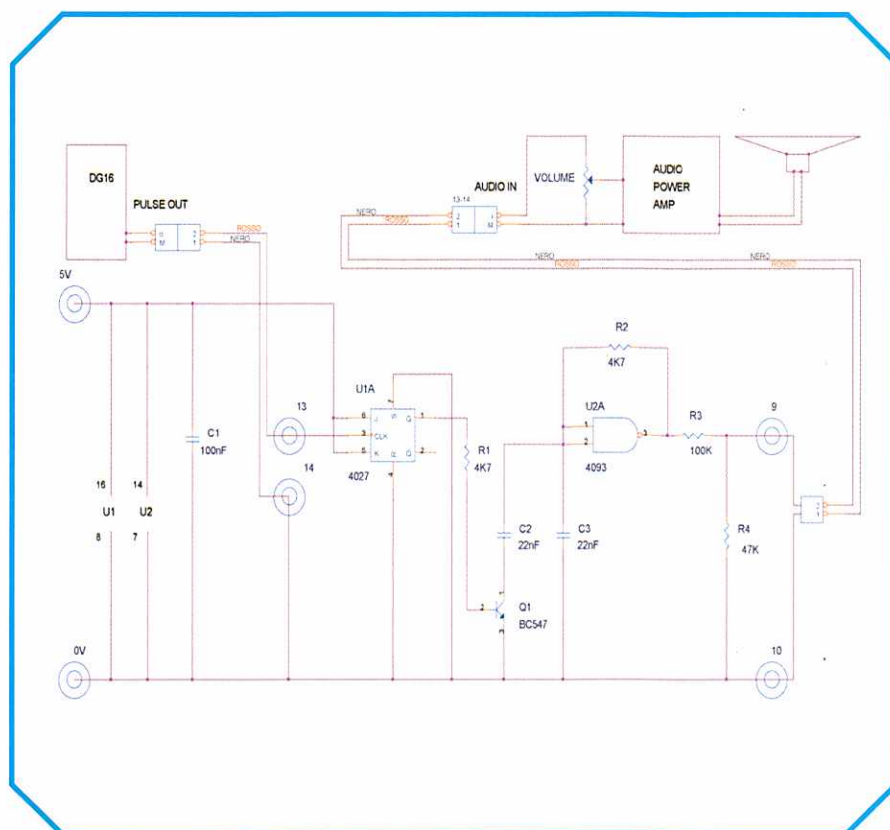
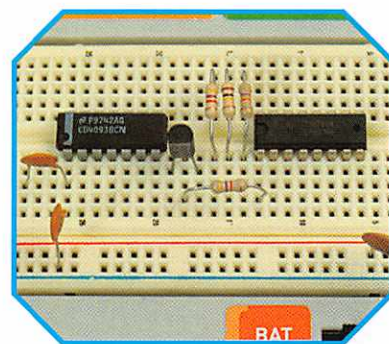
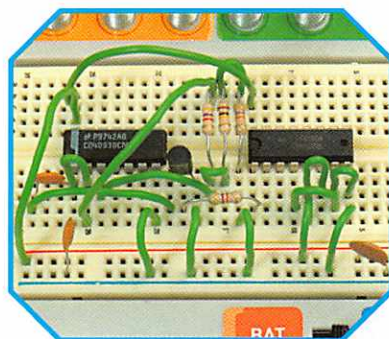
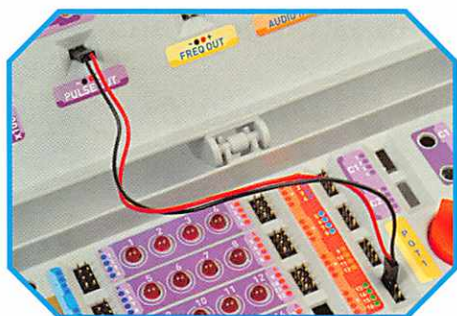
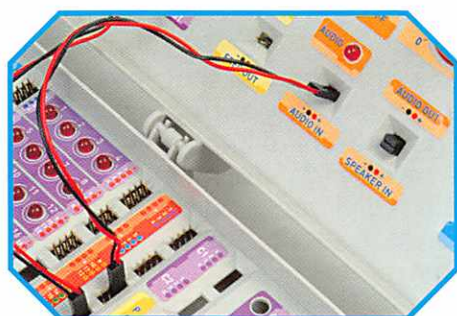
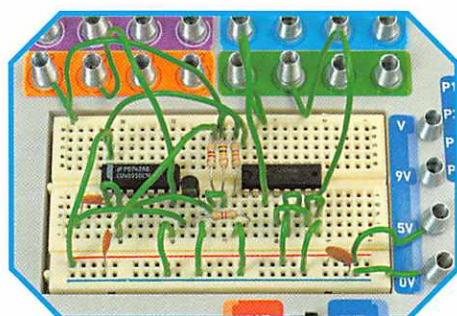
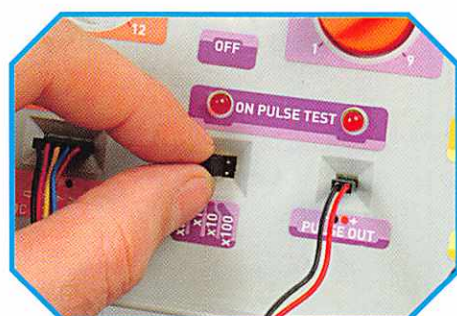
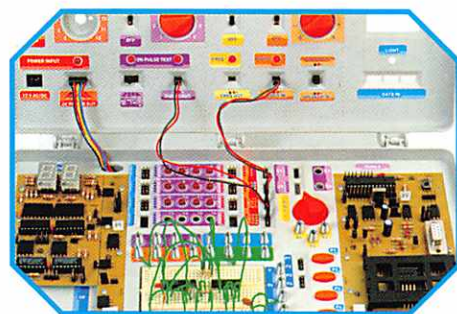
Questo circuito genera un segnale a due toni. È basato su un generatore a frequenza fissa dove viene modificato, mediante un altro circuito, uno dei suoi parametri per ottenere così la seconda frequenza.

Il circuito

In realtà il circuito è piuttosto grande, ma utilizzando le parti che sono già montate sul laboratorio sono sufficienti pochi componenti montati sulla scheda Bread Board.

Osserviamo attentamente lo schema per comprenderne il funzionamento.

Il circuito di base è un noto oscillatore astabile formato dalla porta U2A del circuito integrato 4093; quando il transistor Q1 è in interdizione la frequenza di oscillazione dipende dai valori della resistenza R2 e da C3, la frequenza si abbassa quando questo transistor conduce, dato che collega un altro condensatore C2 in parallelo con il precedente. In questo modo si ottengono le due frequenze.

La variazione di frequenza si ottiene utilizzando come segnale di controllo l'impulso prodotto dal generatore di impulsi del laboratorio, che si ottiene dall'uscita PULSE OUT; questo segnale si divide per due con un bistabile tipo T, formato da uno degli astabili del 4027; in questo modo, all'uscita del bistabile, terminale 1 del 4027, si ottiene un segnale con un ciclo di uscita del 50%, ovvero è un segnale impulsato che rimane per un periodo di tempo uguale a livello alto e a livello basso.

Schema elettrico.

Componenti sulla scheda Bread Board.

Cablaggio della scheda.

Collegamenti a PULSE OUT.

Collegamenti a AUDIO IN.

Vista generale del cablaggio.

Cambiando il ponticello si ottengono suoni diversi.

Vista dell'esperimento completato.

Montaggio

Prima di iniziare il montaggio bisogna verificare che tutte le alimentazioni siano scollegate, quindi la cosa più sicura è impostare il commutatore su EXT e scollegare qualsiasi altro alimentatore esterno. Dopo aver posizionato i componenti sulla scheda Bread Board, come d'abitudine, proseguiremo con il collegamento.

Con un cavetto terminato su due connettori si porta il segnale di uscita del generatore di impulsi fino ai connettori corrispondenti alle molle 13 e 14, il filo rosso al 13 e l'altro al 14. Con un cavetto simile portiamo l'uscita del generatore che abbiamo costruito fino all'ingresso audio AUDIO IN, utilizzando a questo scopo i collegamenti corrispondenti alle molle 9 e 10, filo rosso al 9 e filo nero al 10. Sia il connettore AUDIO IN che PULSE OUT hanno i terminali siglati con due punti, uno rosso e l'altro nero, che devono coincidere con il colore dei fili. Sull'uscita di AUDIO OUT e SPEAKER avremo due ponticelli collegati in posizione orizzontale.

Il ponticello di selezione della banda del generatore di impulsi verrà impostato su X1, anche se successivamente potremo fare delle prove su altre posizioni.

Il comando del volume deve essere vicino al minimo e quello del generatore di impulsi circa a metà della sua corsa.

Prova

Dopo aver verificato che tutto il montaggio sia stato correttamente realizzato si collega l'alimentazione al laboratorio e si alimentano i circuiti ausiliari utilizzati con PULSE ON e AUDIO ON. Si deve poter ascoltare un suono sull'altoparlante che cambierà agendo sul comando FREQUENCY.

LISTA DEI COMPONENTI

U1	Circuito integrato 4027
U2	Circuito integrato 4093
R1,R2	Resistenza 4K7 (giallo, viola, rosso)
R3	Resistenza 100 K (marrone, nero, giallo)
R4	Resistenza 47 K (giallo, viola, arancio)
C1	Condensatore 100 nF
C2, C3	Condensatore 22 nF

Generatore di effetti sonori

Questo circuito permette di variare molti dei suoi parametri per ottenere alcuni suoni, che ci ricordano i grilli e le cicale tipici delle zone rurali.

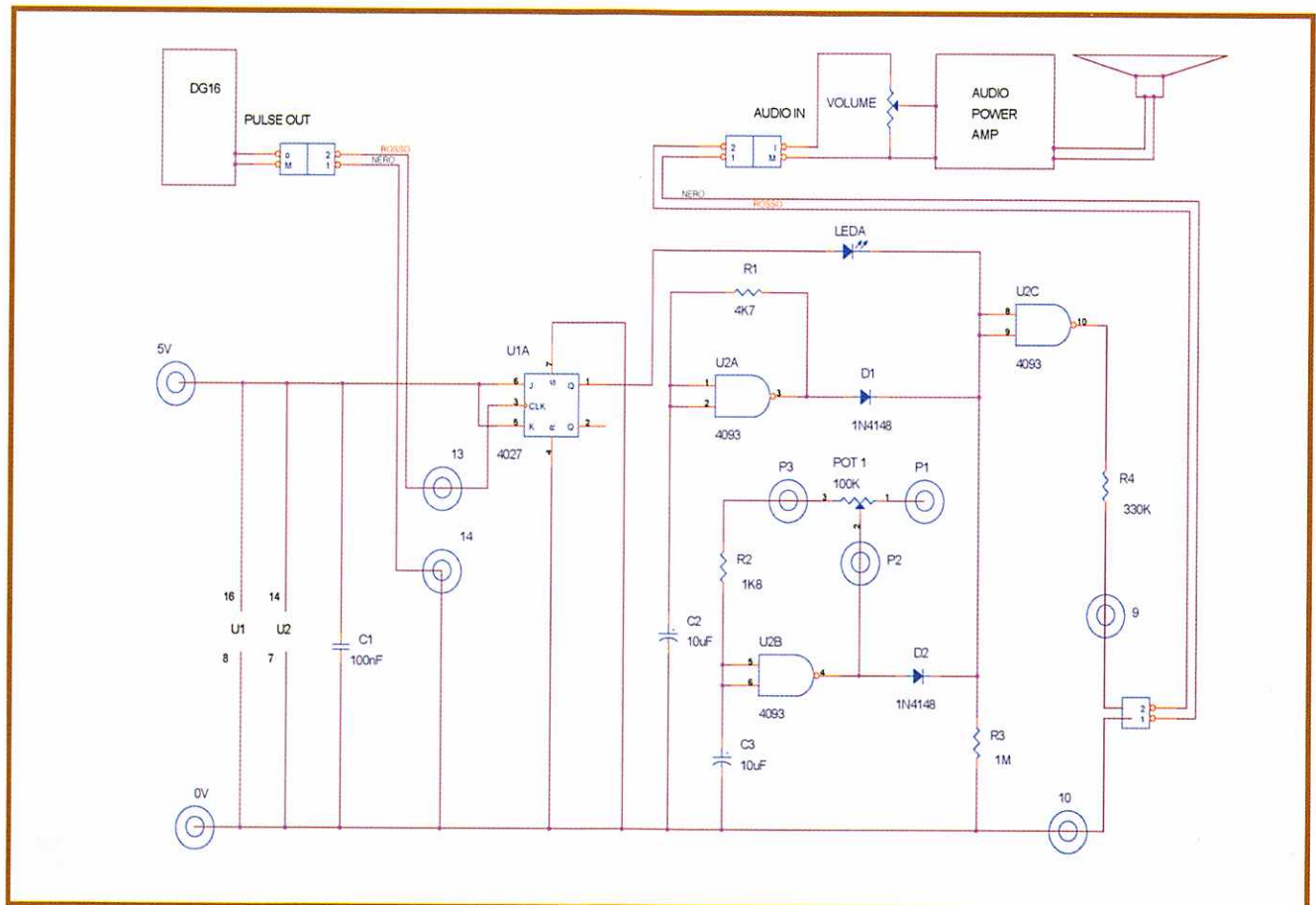
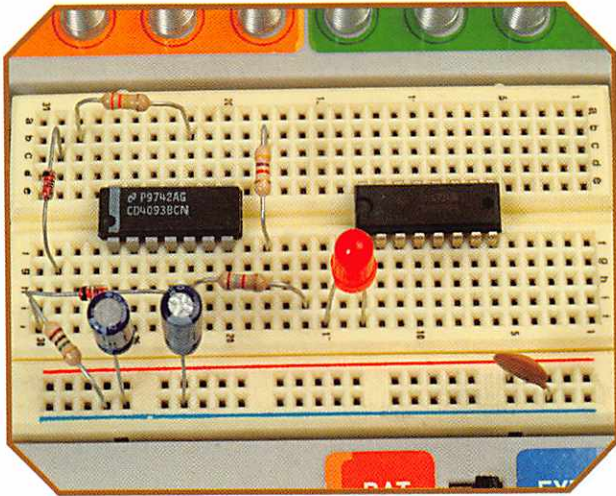
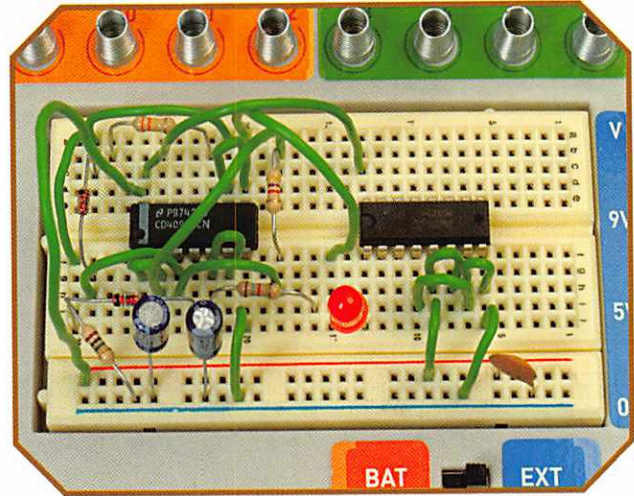
Il circuito

Il circuito è piuttosto originale, utilizza tre generatori di segnale e una porta OR in cui si concentrano i segnali che arrivano dai tre generatori.

Procederemo ora a individuare la posizione di ciascun generatore.

Il più facile da identificare è quello formato dalla porta U2A, a frequenza fissa, la cui frequenza di oscillazione dipende dai valori della resistenza R1 e del condensatore C2, dopo averlo montato, potrete anche provare a cambiare i valori di questi componenti.

Il secondo generatore è simile al primo, è formato attorno alla porta U2B e la sua frequenza di uscita può essere modificata ruotando la manopola di POT1.

Il terzo oscillatore è formato dal generatore di impulsi e da un divisore per due, per ottenere un segnale con il 50% del ciclo di lavoro; con questo sistema è possibile ottenere una vasta gamma di frequenze ruotando la manopola FREQUENCY del generatore di impulsi.

Schema elettrico.

Componenti sulla scheda Bread Board.

Cablaggio interno della scheda.

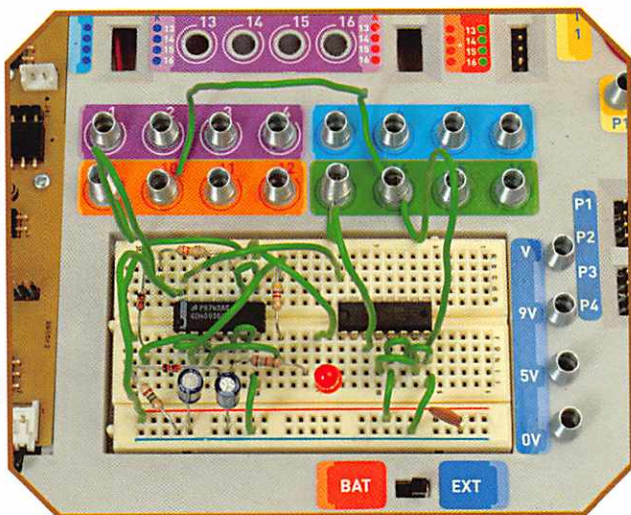
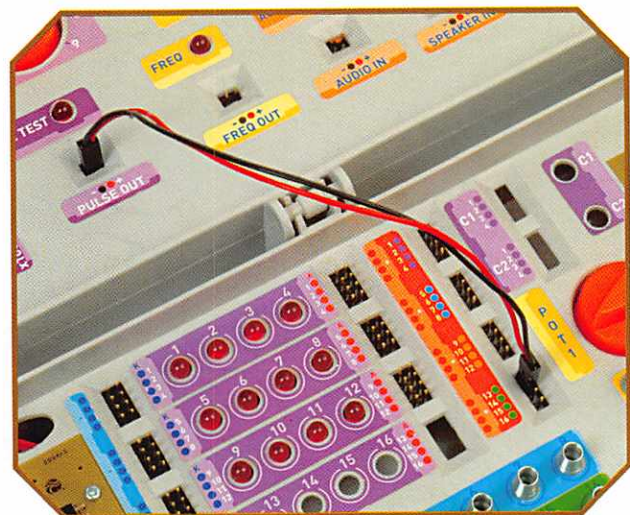
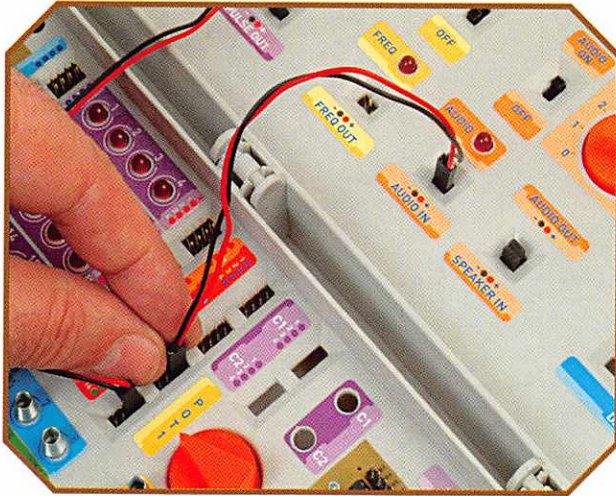
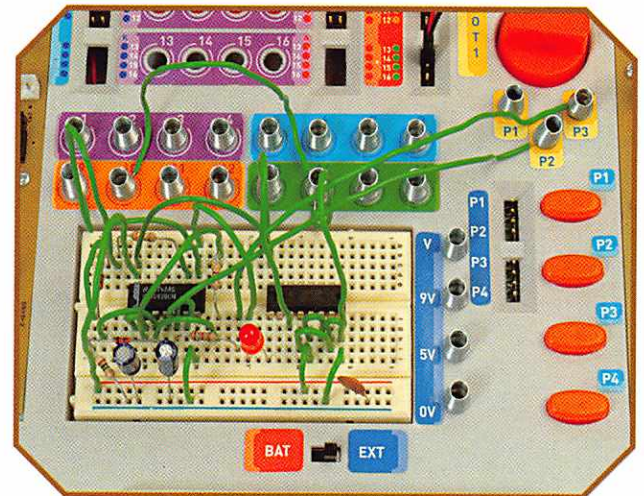
Porta OR

Se osservi,amo lo schema vedremo che l'uscita di ogni generatore è collegata all'anodo di un diodo, mentre i catodi dei tre diodi sono uniti tra loro, in questo modo, si ottiene una porta OR. Il LED, in questo caso, si utilizza come un semplice diodo di segnale, ma dato che su di esso viene fatta circolare una corrente molto bassa, non si illumina. Il punto di unione dei tre LED si porta all'ingresso di una porta NAND che ha i suoi due ingressi uniti, quindi aggiungendo quest'ultima porta invertente alla porta OR precedente otteniamo una porta NOR.

La resistenza R4 abbassa il livello del segnale di uscita applicato all'amplificatore audio.

Montaggio

Dopo aver scelto i componenti necessari si montano sulla scheda Bread Board e così si può iniziare il cablaggio degli stessi seguendo lo schema.

Con due cavetti a due fili si eseguono i collegamenti a PULSE OUT e a AUDIO IN rispettivamente, tenendo presente il colore di ogni conduttore, in quanto non è possibile cambiare il nero con il rosso. I collegamenti del

Collegamenti alle molle.

Collegamenti al generatore di impulsi.

Collegamento dell'audio all'amplificatore.

Dettaglio del collegamento alla scheda Bread Board.

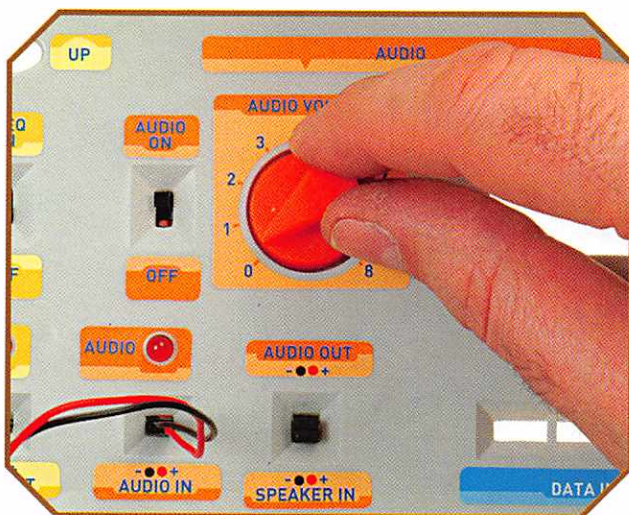
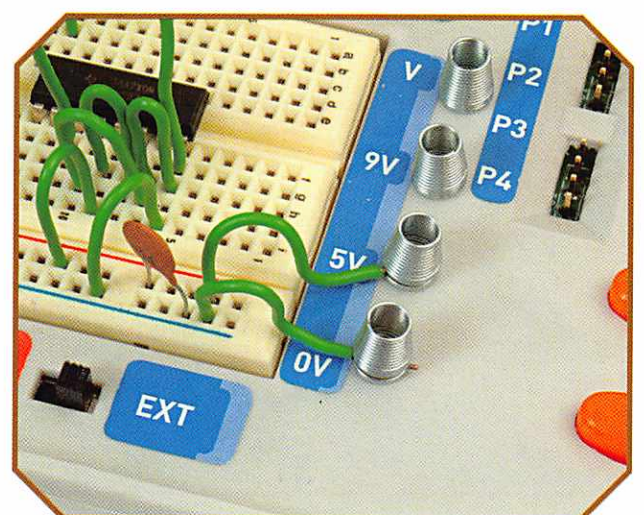
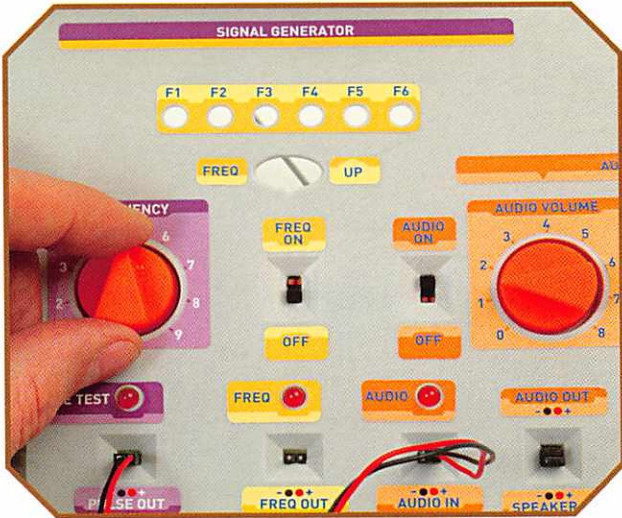
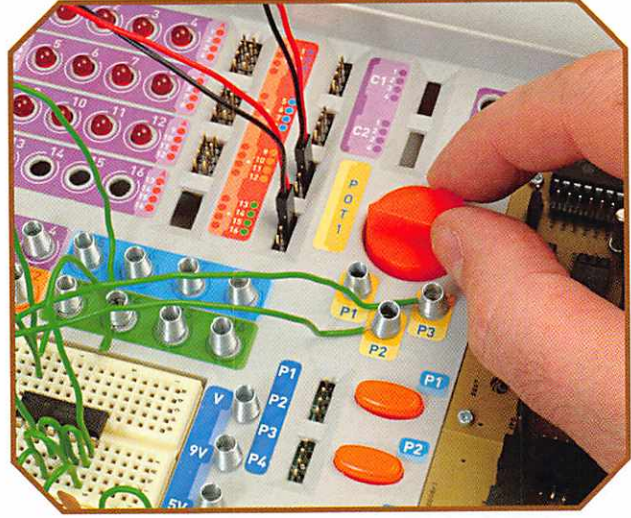
potenziometro POT 1 si realizzano direttamente sulle molle. È altresì necessario collegare le alimentazioni ai due circuiti integrati e verificare che siano inseriti i ponticelli sulle uscite dell'amplificatore AUDIO OUT e sull'altoparlante.

Dopo aver verificato che ogni componente sia al suo posto e che tutti i collegamenti siano corretti si può collegare l'alimentazione.

Funzionamento

Questo circuito può generare una grande quantità di suoni diversi. Vi consigliamo di iniziare posizionando la manopola del volume vicino al minimo, quella di FREQUENCY tra le posizioni 8 e 9, e quella di POT circa a metà della sua corsa.

Collegheremo l'alimentazione del laboratorio a un alimentatore esterno o inseriremo le batterie spostando il commutatore su BAT; sono sufficienti le pile del primo portabatterie, dato che si lavora a 5 volt (4,5 se sono batterie).

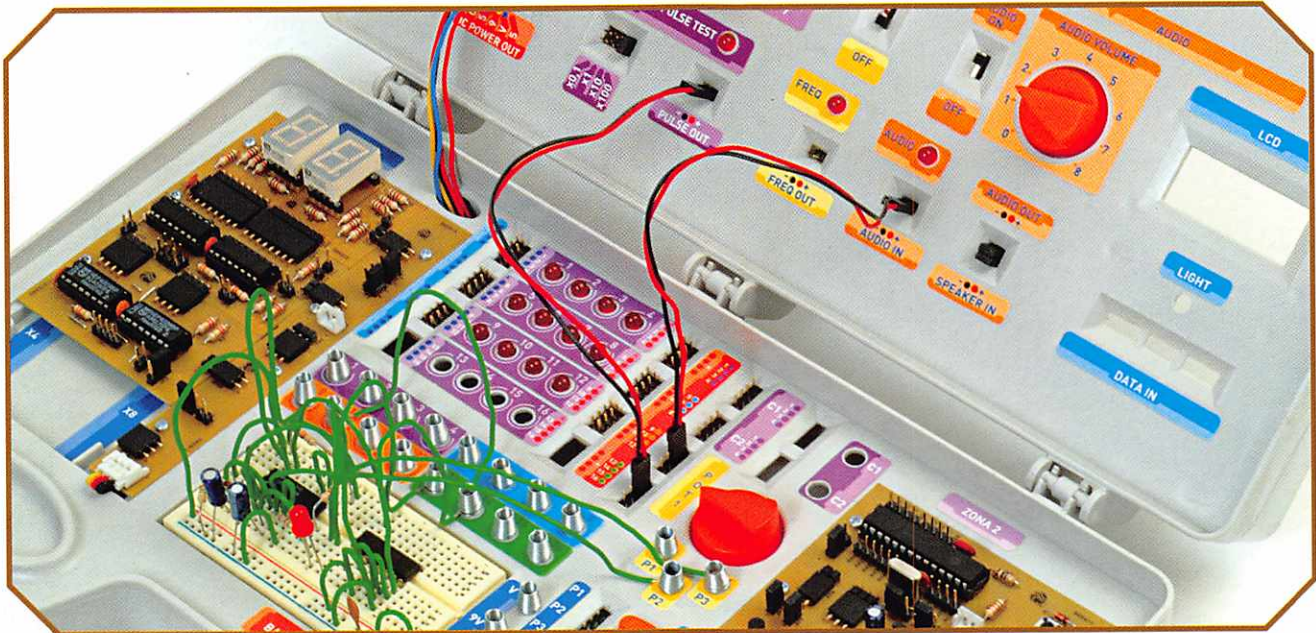
L'amplificatore si collega al commutatore AUDIO ON e il generatore di impulsi con PULSE OUT, in questo modo alimenteremo tutti i circuiti utilizzati. Ruotando la manopola di FREQUENCY e quella di POT 1 otterremo diversi effetti sonori.

Comando del volume al minimo.

Tutto l'esperimento si alimenta a 5 V.

Tra le posizioni 5 e 9 si hanno gli effetti più evidenti.

Con POT1 si può variare anche il suono.

Modifiche

Benché sia possibile ottenere una grande varietà di suoni, girando le manopole POT 1 e FREQUENCY possiamo aumentare ulteriormente le possibilità.

Se si vogliono cambiare i valori di R1 e di C2, che agiscono sul terzo oscillatore, vi consigliamo di provare per R1 con valori minimi da 1K8 fino a 10 K.

LISTA DEI COMPONENTI

U1	Circuito integrato 4027
U2	Circuito integrato 4093
R1	Resistenza 4K7 (giallo, viola, rosso)
R2	Resistenza 1K8 (marrone, grigio, rosso)
R3	Resistenza 1 M (marrone, nero, verde)
R4	Resistenza 330 K (arancio, arancio, giallo)
C1	Condensatore 100 nF
C2,C3	Condensatore 10 µF elettrolitico
D1, D2	Diodo 1N4148
LEDA	Diodo LED rosso 5 mm

Vista dell'esperimento completato.

Esercizio: generatore di onda, il programma

Le applicazioni che si possono sviluppare basandosi sull'utilizzo di un microcontroller, sono molte e diverse tra loro.
In questo caso progettiamo lo sviluppo di un generatore di onda quadra la cui frequenza possa essere determinata dall'utente.

Enunciato

L'obiettivo è quello di progettare un generatore di onda quadra la cui frequenza possa essere determinata dall'utente. Immaginate quali applicazioni possa avere questo generatore: tergicristallo a frequenza variabile di un'automobile, segnale acustico per passaggio pedonale, ecc.

Per determinare la frequenza alla quale verrà generata l'onda utilizzeremo tre ingressi del micro RA2:RA0. In questo modo potremo avere le otto combinazioni della tabella nella figura.

L'onda quadra si otterrà sul terminale di uscita RB0.

Organigramma

La prima cosa da fare prima di progettare un programma che risponda ai requisiti dell'enunciato è plasmarne il funzionamento previsto su un organigramma. Questo ci servirà per sviluppare il codice in modo ordinato e semplice. Nella figura della pagina successiva possiamo vedere l'organigramma generale dell'applicazione, composto da un organigramma per il programma principale e da un altro per la subroutine di servizio all'interrupt. Utilizzando il temporizzatore TMR0 possiamo eseguire una serie di ordini dopo che è trascorso un determinato periodo di tempo o, in altre parole, a una determinata frequenza. Quando il temporizzatore va in overflow si genera un interrupt, ed è nella subroutine dedicata a esso che eseguiremo le azioni pertinenti.

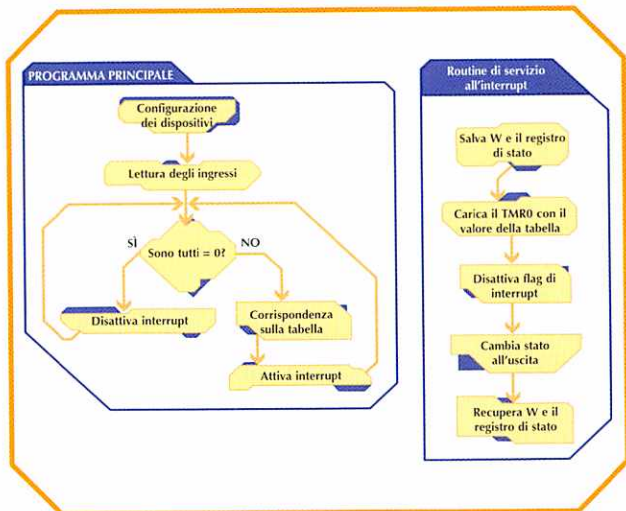
Codice

Intesteremo il codice inserendo sotto forma di commenti ciò che vogliamo che il programma esegua. Continueremo, come sempre, definendo il dispositivo, la libreria in cui si definiscono i registri interni e organizzando la memoria di programma mediante le direttive ORG. Ricordate che in questo programma lavoreremo con gli interrupt, quindi sarà necessario definire il vector di interrupt.

L'altra cosa da fare sarà quella di creare una tabella che contenga i diversi valori da caricare per ottenere le frequenze desiderate. Ogni volta che entreremo in questa subroutine ci sposteremo sulla tabella in funzione dei valo-

Onde quadre di diverse frequenze.

RA2	RA1	RA0	Frequenza	Periodo	Semiperiodo
0	0	0	0 kHz	-- --	-- --
0	0	1	1 kHz	1.000 µs	500 µs
0	1	0	2 kHz	500 µs	250 µs
0	1	1	3 kHz	333 µs	166 µs
1	0	0	4 kHz	250 µs	125 µs
1	0	1	5 kHz	200 µs	100 µs
1	1	0	6 kHz	166 µs	83 µs
1	1	1	7 kHz	143 µs	71 µs

Frequenze in funzione della combinazione degli ingressi.

Organigramma dell'applicazione.

ri degli ingressi e otterremo il valore corrispondente da caricare sul TMR0. Partendo da una frequenza generale di 4 MHz, il TMR0 evolve ogni 1 µs. Selezioneremo un predivisore di 4. Il valore da caricare sul TMR0 si ottiene dividendo il semiperiodo della frequenza desiderata con il predivisore. Al valore ottenuto sottrarremo 2, per motivi di sincronismo interno del PIC, convertiremo in esadecimale ed eseguiremo il complemento. Fatto questo, intesteremo il programma principale con l'etichetta inizio e configureremo i dispositivi con cui vogliamo lavorare. Configureremo la porta B come uscita. Programmeremo il registro ADCON1 per fare in modo che la porta A sia a segnali digitali e il registro TRISA per fare in modo che questi segnali siano ingressi. Nel registro OPTION_REG configureremo il predivisore del temporizzatore e in INT-CON abiliteremo l'interrupt per overflow del TMR0.

Per realizzare il generatore dobbiamo leggere il valore della porta di ingresso. Se i tre

```
GeneratoreOnda - Blocco note
File  Modifica  Formato  Visualizza  ?

;**************************************************************************
;Tabella: questa routine restituisce il valore da caricare sul TMR0 in base alla frequenza sele-
;zionata. Partendo da una frequenza generale di 4 MHz, il TMR0 evolve ogni 1 µs.
;Si seleziona un prescaler da 4. Il valore da caricare sul TMR0 si ottiere dividendo il
;semiperiodo della frequenza desiderata con il prescaler. Al valore otteruto si
;sottrae 2, per motivi di sincronismo interno del PIC; si converte in hex. e si comple-
;menta.

Tabella:        addwf   PCL,F           ;Calcola lo spostamento della tabella
                retlw   0x00            ;0 KHz
                retlw   0x86            ;1 KHz
                retlw   0xc5            ;2 KHz
                retlw   0xda            ;3 KHz
                retlw   0xe4            ;4 KHz
                retlw   0xea            ;5 KHz
                retlw   0xee            ;6 KHz
                retlw   0xf1            ;7 KHz
```

Tabella per le diverse frequenze.

```
GeneratoreOnda - Blocco note
File  Modifica  Formato  Visualizza  ?
;       ESERCIZIO: Generazione di un'onda quadra a diverse frequenze variando il valore del TMR0
;       -----------------------------------------------------------------------------------
;
;La linea di uscita RB0 cambierà di stato ad una frequenza determinata dal valore
;caricato mediante i 3 interruttori RA0-RA2:
;
;RA2    RA1    RA0    Frequenza     Periodo      Semiperiodo
;---    ---    ---    ----------    -------      -----------
;0      0      0      0 KHZ         ---          ---
;0      0      1      1 KHZ         1000 µs      500 µs
;0      1      0      2 KHZ         500 µs       250 µs
;0      1      1      3 KHZ         333 µs       166 µs
;1      0      0      4 KHZ         250 µs       125 µs
;1      0      1      5 KHZ         200 µs       100 µs
;1      1      0      6 KHZ         166 µs       83 µs
;1      1      1      7 KHZ         143 µs       71 µs
;
;Nel trattamento dell'interrupt provocato dall'overflow del TMR0, notiamo
;il salvataggio del registro W e del registro di stato, per recuperarli successivamente.
;Questa operazione è definita "salvataggio del contesto".
;

                List    p=16F870        ;Tipo di processore
                include "P16F870.INC"   ;Definizioni dei registri interni

                org     0x00            ;Vector di Reset
                goto    Inizio
                org     0x04            ;Vector di interrupt
                goto    Interrupt

;**************************************************************************
```

Intestazione del codice.

ingressi sono a zero non si genera nessuna onda, si disabilitano gli interrupt e si salta nuovamente alla lettura degli ingressi. Se al contrario, uno o più ingressi sono a 1, il programma chiamerà la tabella per ottenere il valore corrispondente da caricare sul TMR0. Questo valore lo caricheremo su una variabile che sarà stata definita in precedenza. Abiliteremo gli interrupt e salteremo nuovamente all'inizio del ciclo per leggere di nuovo gli ingressi. Mentre si esegue questo ciclo il temporizzatore continua la sua corsa e quando va in overflow provoca un interrupt.

Nella subroutine di servizio all'interrupt la prima cosa da fare è salvare l'ambiente di lavoro.

Salvare l'ambiente di lavoro o contesto consiste nel salvare il contenuto dei registri di lavoro e di stato per poi poterlo recuperare quando usciremo dalla subroutine. Sarà necessario, quindi, creare tre nuove variabili per contenere temporaneamente questi registri.

Successivamente scaricheremo il valore del temporizzatore sul suo registro per reinizializzarlo e attiveremo nuovamente l'interrupt resettando il bit del flag.

```
GeneratoreOnda - Blocco note
File  Modifica  Formato  Visualizza  ?

Inizio      clrf    PORTB           ;Cancella i latch di uscita
            bsf     STATUS,RP0      ;Seleziona banco 1
            clrf    TRISB           ;Porta B si configura come uscita
            movlw   b'00000110'
            movwf   ADCON1          ;Porta A digitale
            movlw   b'00011111'
            movwf   TRISA           ;RA0-RA4 ingressi
            movlw   b'00000001'
            movwf   OPTION_REG      ;Prescaler di 4 per il TMR0
            bcf     STATUS,RP0      ;Seleziona banco 0
            movlw   b'00100000'
            movwf   INTCON          ;Interrupt TMR0 abilitato
```

Configurazione dei dispositivi.

GeneratoreOnda - Blocco note

File Modifica Formato Visualizza ?

```
Loop            clrwdt                  ;Aggiorna il WDT
                movf    PORTA,W
                andlw   b'00000111'
                btfss   STATUS,Z        ;RA0-RA2 = 0?
                goto    Uscita_On       ;No, uscita del segnale in frequenza
                bcf     INTCON,GIE      ;Si, Interrupt OFF, frequenza OFF
                goto    Loop

Uscita_On       call    Tabella         ;Determina il valore da caricare sul TMR0
                movwf   Valore          ;Carica la variabile
                bsf     INTCON,GIE      ;Interrupt ON
                goto    Loop

                end                     ;Fine del programma sorgente
```

Programma principale.

Agiremo sull'uscita cambiandone il valore, ovvero imposteremo il valore opposto a quello che aveva in precedenza. Questo lo possiamo fare mediante l'istruzione xorwf. Infine non ci rimane che recuperare il contesto, ovvero i registri di lavoro e di stato, e concludere la subroutine.

Compilazione

Il file che vi abbiamo spiegato lo potete trovare sul secondo CD allegato all'opera all'interno della cartella "Varie" con il nome "GeneratoreOnda.asm". Questo file, o quello che voi avete creato seguendo le spiegazioni, deve essere compilato e simulato.

Apriamo MPLAB e creiamo un nuovo progetto, ad esempio con il nome "onda.pjt". Aggiungiamo il nostro codice editando il progetto e selezionando Add Node.

Dato che il nome del file è superiore agli 8 caratteri, quando lo selezioneremo apparirà come "GENERA~1.asm". Visualizzeremo a monitor il progetto aprendo il file e procederemo alla compilazione.

Il codice si compila e si assembla con successo generando solamente tre messaggi inerenti ai banchi di memoria.

GeneratoreOnda - Blocco note

File Modifica Formato Visualizza ?

```
Interrupt       movwf   W_Temp          ;Salva il registro W
                swapf   STATUS,W
                movwf   Status_Temp     ;Salva il registro di stato

                movf    Valore,W
                movwf   TMR0            ;Ricarica il TMR0
                bcf     INTCON,T0IF     ;Disattiva il Flag TMR0
                movlw   b'00000001'
                xorwf   PORTB,F         ;Toggle su RB0

                swapf   Status_Temp,W
                movwf   STATUS          ;Recupera il registro di stato
                swapf   W_Temp,F
                swapf   W_Temp,W        ;Recupera il registro W

                retfie
```

Subroutine di servizio all'interrupt.

Simulazione

Apriamo le finestre abituali: Registri delle Funzioni Speciali e quella di visualizzazione dei registri più importanti. In quest'ultima visualizzeremo PORTB, PORTA e w in binario, la variabile Valore in esadecimale.

Simuleremo una prima esecuzione senza aver attivato i terminali degli ingressi. In questo modo, quando il programma legge lo stato degli ingressi e vede che sono tutti a 0, disattiverà gli interrupt e tornerà a ripetere il ciclo, esattamente come volevamo.

Per ingannare il programma e simulare degli ingressi attivi utilizzeremo la finestra Modify. Quando l'esecuzione arriva alla linea "btfss STATUS,Z", sceglieremo sul campo Address il bit Z e sul campo Data/Optcode gli assegneremo il valore 1 in decimale. Selezioneremo Write per modificare il bit e verifichere-

GeneratoreOnda - Blocco note

File Modifica Formato Visualizza ?

```
                List    p=16F870        ;Tipo di processore
                include "P16F870.INC"   ;Definizioni dei registri interni

Valore          equ     0x20            ;Variabile della frequenza
W_Temp          equ     0x21            ;W temporale
Status_Temp     equ     0x22            ;Registro di stato temporale

                org     0x00            ;Vector di Reset
                goto    Inizio
                org     0x04            ;Vector di interrupt
                goto    Interrupt

;*****************************************************************************************
```

Definiamo le variabili all'inizio del programma.

```
Build Results
Building GENERA~1.HEX...

Compiling GENERA~1.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\GENERA~1.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\GENERA~1.ASM 73 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\GENERA~1.ASM 75 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\GENERA~1.ASM 77 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\GENERA~1.ASM 79 : Register in operand not in bank 0.  Ensure

Build completed successfully.
```

Risultato della compilazione.

mo come, eseguendo la linea successiva, salteremo direttamente alla routine di interrupt.

Una volta all'interno della routine, faremo la stessa operazione, ma questa volta con la variabile Valore a cui assegneremo un valore qualsiasi tra quelli della tabella (in esadecimale), ad esempio "ea" (5 kHz). Vedremo come vengono assegnati correttamente i valori ai registri e come la porta B di uscita cambia stato.

All'uscita della routine di interrupt il simulatore emette un messaggio di allarme che si può risolvere utilizzando punti di arresto o Break points durante la simulazione.

Provate diverse alternative di simulazione tenendo presente che tutte le simulazioni che comprendono temporizzatori e interrupt sono molto complicate.

È importante che nella simulazione si verifichi il corretto andamento del programma per controllare che non ci siano errori di funzionamento, ma quando si utilizzano i temporizzatori non è possibile sapere se il programma funzionerà in modo adeguato fino a quando non si realizza il montaggio elettrico.

MPLAB IDE - C:\PROGRA~1\MPLAB\PROGETTI\ONDA.PJT

File Project Edit Debug PICSTART Plus Options Tools Window Help

c:\progra~1\mplab\progetti\genera~1.asm

```
Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Porta B si configura como uscita
                movlw   b'00000110'
                movwf   ADCON1          ;Porta A digitale
                movlw   b'00011111'
                movwf   TRISA           ;RA0-RA4 ingressi
                movlw   b'00000001'
                movwf   OPTION_REG      ;Prescaler di 4 per il TMR0
                bcf     STATUS,RP0      ;Seleziona banco 0
                movlw   b'00100000'
                movwf   INTCON          ;Interrupt TMR0 abilitato

Loop            clrwdt                  ;Aggiorna il WDT
                movf    PORTA,W
                andlw   b'00000111'
                btfss   STATUS,Z        ;RA0-RA2 = 0?
                goto    Uscita_On       ;No, uscita del segnale in frequenza
                bcf     INTCON,GIE      ;Si, Interrupt OFF, frequenza OFF
                goto    Loop

Uscita_On       call    Tabella         ;Determina il Valore da caricare sul TMR0
                movwf   Valore          ;Carica la variabile
                bsf     INTCON,GIE      ;Interrupt ON
                goto    Loop

                end                     ;Fine del programma sorgente
```

Special Function Register Wi...

SFR Name	Hex	Dec	Binary
w	00	0	00000000
tmr0	15	21	00010101
option_reg	01	1	00000001
pcl	2A	42	00101010
pclath	00	0	00000000
status	1F	31	00011111
fsr	00	0	00000000
porta	00	0	00000000
trisa	1F	31	00011111
portb	00	0	00000000
trisb	00	0	00000000
portc	00	0	00000000
trisc	FF	255	11111111
intcon	20	32	00100000
pir1	00	0	00000000
pie1	00	0	00000000
pir2	00	0	00000000
pie2	00	0	00000000
tmr1l	00	0	00000000
tmr1h	00	0	00000000
t1con	00	0	00000000
tmr2	00	0	00000000
pr2	FF	255	11111111
t2con	00	0	00000000
ccpr1l	00	0	00000000
ccpr1h	00	0	00000000
ccp1con	00	0	00000000
rcsta	00	0	00000000
txreg	00	0	00000000
rcreg	00	0	00000000
txsta	02	2	00000010
spbrg	00	0	00000000
adresh	00	0	00000000
adresl	00	0	00000000
adcon0	00	0	00000000
adcon1	06	6	00000110
pcon	00	0	00000000
eedata	00	0	00000000

Watch_1

Address	Symbol	Value
06	PORTB	B'00000000'
05	PORTA	B'00000000'
00	W	B'00000000'
20	Valore	H'EA'

Modify

Address: Z — End Address: — Data/Opcode: 1 — Fill Range — Read Range — Radix: Hex / Decimal — Auto Increment — Memory Area: Data / Program / Stack / EEPROM — Write — Read — Close — Help

Ln 87 Col 1 | 98 | WR | No Wrap | INS | PIC16F870 | pc:0x2a | w:0x00 | --Z DC C | Bk On | Sim | 4 MHz | Proj

Aspetto di MPLAB durante la simulazione.